

Peer-review of assignment 4 for *INF3331-erlendwe*

Reviewer 1, UiO-pederhot, pederhot@uio.no

October 16, 2016

1 Introduction

1.1 Goal

The review should provide feedback on the solution to the student. The main goal is to *give constructive feedback and advice* on how to improve the solution. You, the peer-review team, can decide how you organise the peer-review work between you.

1.2 Guidelines

For each (coding) exercise, one should review the following points:

- Is the code **working as expected**? For non-internal functions (in particular for scripts that are run from the command-line), does the program handle invalid inputs sensibly?
- Is the code **well documented**? Are there docstrings and are they useful?
- Is the code written in **Pythonic way**¹? Is the code easy to read? Are the variable/class/function names sensible? Do you find overuse of classes or not sufficient use of functions or classes? Are there parts of the program that are hard to understand?
- Can you find **unnecessarily complicated parts** of the program? If so, suggest an improved implementation.
- List the programming parts that are not answered.

Use (shortened) code snippets where appropriate to show how to improve the solution.

1.3 Points

The review is completed by pushing the review Latex and PDF files to each of the reviewed repositories. The name of the files should be *feedback.tex* and *feedback.pdf*.

You will get up to 10 points for delivering the peer-reviews. Each of you should contribute to the review roughly equivalently - your team will get the same number of points².

2 Review

For this review I've used Python 3.5.2, and a broken numpy that I've unable to fix.

General feedback

I believe you've done a great job on this assignment, assuming that the files I could not run work the way they should! I had a hard time locating any documentation/commenting in the code. You have structured your code in a good way, so it's not hard to read. Could not locate any submissions for 4.7 or 4.8. Also not looking for a 4.4 submission because the repository is INF3331.

Assignment 4.1: Python implementation

Your native implementation works well! You had a progressbar implemented in the code, but this required extra modules that I did not have. Next time do provide an easy way for me to install that, so I can go on about the review without messing with new modules, as this was supposed to be "native" python. Worry not though, I got the code to work after commenting out the progressbar. Other than this, a few comments here and there wouldn't hurt. The code looks clean and nice, well done!

¹<https://www.python.org/dev/peps/pep-0020/>

²In case a team-member does not contribute, please email simon@simula.no

Assignment 4.2: numpy implementation

Your numpy implementation works well, like your native one. I see that it is a bit slower than the native one, and then I stumbled upon your report on the implementation. In `mandelbrot_2.py` you have two for-loops (one linear and one exponential, and this might be the cause to why your numpy implementation is taking longer, like you've said in your report.

- I don't see any sign of vectorization in your code, I might be blind though

Assignment 4.3: Integrated C implementation

I'm not able to compile the code, as the setup gives me a numpy warning that I'm not able to solve.

Assignment 4.5: User interface

I try running `python3 mandelbrot_cli.py` in my console, expecting it to ask me for input. It seems to start running some other code, that I cannot locate. I might be blind again, but this does not work! Please include a set of instructions next time, if it's not intuitive to run.

Assignment 4.6: Packaging and unit tests

No issues with the setup! I'm having some serious numpy problems, and am therefore not able to run the unit test on my laptop. But the unit test file does look like it would work, if I had the chance to run it. Sorry!

References

[Gre93] George D. Greenwade. The Comprehensive Tex Archive Network (CTAN). *TUGBoat*, 14(3):342–351, 1993.